SMART ARM-based Microcontrollers

AT03261: SAM D/R/L/C System Interrupt (SYSTEM INTERRUPT) Driver

APPLICATION NOTE

Introduction

This driver for Atmel® | SMART ARM®-based microcontrollers provides an interface for the configuration and management of internal software and hardware interrupts/exceptions.

The following peripheral is used by this module:

- NVIC (Nested Vector Interrupt Controller)

The following devices can use this module:

- Atmel | SMART SAM D20/D21
- Atmel | SMART SAM R21
- Atmel | SMART SAM D09/D10/D11
- Atmel | SMART SAM L21/L22
- Atmel | SMART SAM DA1
- Atmel | SMART SAM C20/C21

The outline of this documentation is as follows:

Table of Contents

1. Software License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.

4. This software may only be redistributed and used in connection with an Atmel microcontroller product.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Prerequisites

There are no prerequisites for this module.

3. Module Overview

The ARM® Cortex® M0+ core contains an interrupt and exception vector table, which can be used to configure the device's interrupt handlers; individual interrupts and exceptions can be enabled and disabled, as well as configured with a variable priority.

This driver provides a set of wrappers around the core interrupt functions, to expose a simple API for the management of global and individual interrupts within the device.

3.1. Critical Sections

In some applications it is important to ensure that no interrupts may be executed by the system whilst a critical portion of code is being run; for example, a buffer may be copied from one context to another - during which interrupts must be disabled to avoid corruption of the source buffer contents until the copy has completed. This driver provides a basic API to enter and exit nested critical sections, so that global interrupts can be kept disabled for as long as necessary to complete a critical application code section.

3.2. Software Interrupts

For some applications, it may be desirable to raise a module or core interrupt via software. For this reason, a set of APIs to set an interrupt or exception as pending are provided to the user application.

4. Special Considerations

Interrupts from peripherals in the SAM devices are on a per-module basis; an interrupt raised from any source within a module will cause a single, module-common handler to execute. It is the user application or driver's responsibility to de-multiplex the module-common interrupt to determine the exact interrupt cause.

5. Extra Information

For extra information, see Extra Information for SYSTEM INTERRUPT Driver. This includes:

- Acronyms
- Dependencies
- Errata
- Module History

6. Examples

For a list of examples related to this driver, see Examples for SYSTEM INTERRUPT Driver.

7. API Overview

7.1. Function Definitions

7.1.1. Critical Section Management

7.1.1.1. Function system_interrupt_enter_critical_section()

Enters a critical section.

```
void system_interrupt_enter_critical_section( void )
```

Disables global interrupts. To support nested critical sections, an internal count of the critical section nesting will be kept, so that global interrupts are only re-enabled upon leaving the outermost nested critical section.

7.1.1.2. Function system_interrupt_leave_critical_section()

Leaves a critical section.

```
void system_interrupt_leave_critical_section( void )
```

Enables global interrupts. To support nested critical sections, an internal count of the critical section nesting will be kept, so that global interrupts are only re-enabled upon leaving the outermost nested critical section.

7.1.2. Interrupt Enabling/Disabling

7.1.2.1. Function system_interrupt_is_global_enabled()

Check if global interrupts are enabled.

```
bool system_interrupt_is_global_enabled( void )
```

Checks if global interrupts are currently enabled.

Returns
A boolean that identifies if the global interrupts are enabled or not.

Table 7-1. Return Values

Return value	Description
true	Global interrupts are currently enabled
false	Global interrupts are currently disabled

7.1.2.2. Function system_interrupt_enable_global()

Enables global interrupts.

```
void system_interrupt_enable_global( void )
```

Enables global interrupts in the device to fire any enabled interrupt handlers.

7.1.2.3. Function system_interrupt_disable_global()

Disables global interrupts.

```
void system_interrupt_disable_global( void )
```

Disabled global interrupts in the device, preventing any enabled interrupt handlers from executing.

7.1.2.4. Function system_interrupt_is_enabled()

Checks if an interrupt vector is enabled or not.

```
bool system_interrupt_is_enabled(
        const enum system_interrupt_vector vector)
```

Checks if a specific interrupt vector is currently enabled.

Table 7-2. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector number to check

Returns

A variable identifying if the requested interrupt vector is enabled.

Table 7-3. Return Values

Return value	Description
true	Specified interrupt vector is currently enabled
false	Specified interrupt vector is currently disabled

7.1.2.5. Function system_interrupt_enable()

Enable interrupt vector.

```
void system_interrupt_enable(
        const enum system_interrupt_vector vector)
```

Enables execution of the software handler for the requested interrupt vector.

Table 7-4. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector to enable

7.1.2.6. Function system_interrupt_disable()

Disable interrupt vector.

```
void system_interrupt_disable(
        const enum system_interrupt_vector vector)
```

Disables execution of the software handler for the requested interrupt vector.

Table 7-5. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector to disable

7.1.3. Interrupt State Management

7.1.3.1. Function system_interrupt_get_active()

Get active interrupt (if any).

```
enum system_interrupt_vector system_interrupt_get_active( void )
```

Return the vector number for the current executing software handler, if any.

Returns
Interrupt number that is currently executing.

7.1.3.2. Function system_interrupt_is_pending()

Check if a interrupt line is pending.

```
bool system_interrupt_is_pending(
        const enum system_interrupt_vector vector)
```

Checks if the requested interrupt vector is pending.

Table 7-6. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector number to check

Returns
A boolean identifying if the requested interrupt vector is pending.

Table 7-7. Return Values

Return value	Description
true	Specified interrupt vector is pending
false	Specified interrupt vector is not pending

7.1.3.3. Function system_interrupt_set_pending()

Set a interrupt vector as pending.

```
enum status_code system_interrupt_set_pending(
        const enum system_interrupt_vector vector)
```

Set the requested interrupt vector as pending (i.e. issues a software interrupt request for the specified vector). The software handler will be handled (if enabled) in a priority order based on vector number and configured priority settings.

Table 7-8. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector number which is set as pending

Returns
Status code identifying if the vector was successfully set as pending.

Table 7-9. Return Values

Return value	Description
STATUS_OK	If no error was detected
STATUS_INVALID_ARG	If an unsupported interrupt vector number was given

7.1.3.4. Function system_interrupt_clear_pending()

Clear pending interrupt vector.

```
enum status_code system_interrupt_clear_pending(
        const enum system_interrupt_vector vector)
```

Clear a pending interrupt vector, so the software handler is not executed.

Table 7-10. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector number to clear

Returns
A status code identifying if the interrupt pending state was successfully cleared.

Table 7-11. Return Values

Return value	Description
STATUS_OK	If no error was detected
STATUS_INVALID_ARG	If an unsupported interrupt vector number was given

7.1.4. Interrupt Priority Management

7.1.4.1. Function system_interrupt_set_priority()

Set interrupt vector priority level.

```
enum status_code system_interrupt_set_priority(
        const enum system_interrupt_vector vector,
        const enum system_interrupt_priority_level priority_level)
```

Set the priority level of an external interrupt or exception.

Table 7-12. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector to change
[in]	priority_level	New vector priority level to set

Returns
Status code indicating if the priority level of the interrupt was successfully set.

Table 7-13. Return Values

Return value	Description
STATUS_OK	If no error was detected
STATUS_INVALID_ARG	If an unsupported interrupt vector number was given

7.1.4.2. Function system_interrupt_get_priority()

Get interrupt vector priority level.

```
enum system_interrupt_priority_level system_interrupt_get_priority(
        const enum system_interrupt_vector vector)
```

Retrieves the priority level of the requested external interrupt or exception.

Table 7-14. Parameters

Data direction	Parameter name	Description
[in]	vector	Interrupt vector of which the priority level will be read

Returns
Currently configured interrupt priority level of the given interrupt vector.

7.2. Enumeration Definitions

7.2.1. Enum system_interrupt_priority_level

Table of all possible interrupt and exception vector priorities within the device.

Table 7-15. Members

Enum value	Description
SYSTEM_INTERRUPT_PRIORITY_LEVEL_0	Priority level 0, the highest possible interrupt priority
SYSTEM_INTERRUPT_PRIORITY_LEVEL_1	Priority level 1
SYSTEM_INTERRUPT_PRIORITY_LEVEL_2	Priority level 2
SYSTEM_INTERRUPT_PRIORITY_LEVEL_3	Priority level 3, the lowest possible interrupt priority

7.2.2. Enum system_interrupt_vector_samc21

Table of all possible interrupt and exception vector indexes within the SAM C20/C21 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-16. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_SYSTEM	Interrupt vector index for MCLK, OSCCTRL, OSC32KCTRL, PAC, PM, SUPC, TAL peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_FREQM	Interrupt vector index for Frequency Meter peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).

Enum value	Description
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).

7.2.3. Enum system_interrupt_vector_samd09

Table of all possible interrupt and exception vector indexes within the SAM D09 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-17. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt

7.2.4. Enum system_interrupt_vector_samd1x

Table of all possible interrupt and exception vector indexes within the SAM D10/D11 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-18. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_DAC	Interrupt vector index for a Digital-to-Analog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_PTC	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt

7.2.5. Enum system_interrupt_vector_samd20

Table of all possible interrupt and exception vector indexes within the SAM D20 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-19. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).

Enum value	Description
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC0`).
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_DAC	Interrupt vector index for a Digital-to-Analog peripheral interrupt

7.2.6. Enum system_interrupt_vector_samd21

Table of all possible interrupt and exception vector indexes within the SAM D21 device. Check peripherals configuration in SAM D21 datasheet for available vector index for specific device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-20. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_USB	Interrupt vector index for a Universal Serial Bus interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_DAC	Interrupt vector index for a Digital-to-Analog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_PTC	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_I2S	Interrupt vector index for a Inter-IC Sound Interface peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AC1	Interrupt vector index for an Analog Comparator 1 peripheral interrupt

7.2.7. Enum system_interrupt_vector_samda1

Table of all possible interrupt and exception vector indexes within the SAM DA1 device. Check peripherals configuration in SAM DA1 datasheet for available vector index for specific device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-21. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_USB	Interrupt vector index for a Universal Serial Bus interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_DAC	Interrupt vector index for a Digital-to-Analog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_PTC	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt
SYSTEM_INTERRUPT_MODULE_I2S	Interrupt vector index for a Inter-IC Sound Interface peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AC1	Interrupt vector index for an Analog Comparator 1 peripheral interrupt

7.2.8. Enum system_interrupt_vector_saml21

Table of all possible interrupt and exception vector indexes within the SAM L21 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-22. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_SYSTEM	Interrupt vector index for MCLK, OSCCTRL, OSC32KCTRL, PAC, PM, SUPC, TAL peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_USB	Interrupt vector index for a Universal Serial Bus interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).

Enum value	Description
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_DAC	Interrupt vector index for a Digital-to-Analog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_PTC	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AES	Interrupt vector index for a AES peripheral interrupt
SYSTEM_INTERRUPT_MODULE_TRNG	Interrupt vector index for a TRNG peripheral interrupt

7.2.9. Enum system_interrupt_vector_saml22

Table of all possible interrupt and exception vector indexes within the SAM L22 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-23. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_SYSTEM	Interrupt vector index for MCLK, OSCCTRL, OSC32KCTRL, PAC, PM, SUPC, TAL peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt

Enum value	Description
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_FREQM	Interrupt vector index for Frequency Meter peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_TCC0	Interrupt vector index for a Timer/Counter Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SLCD	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt

7.2.10. Enum system_interrupt_vector_samr21

Table of all possible interrupt and exception vector indexes within the SAM R21 device.
Note: The actual enumeration name is "system_interrupt_vector".

Table 7-24. Members

Enum value	Description
SYSTEM_INTERRUPT_NON_MASKABLE	Interrupt vector index for a NMI interrupt
SYSTEM_INTERRUPT_HARD_FAULT	Interrupt vector index for a Hard Fault memory access exception
SYSTEM_INTERRUPT_SV_CALL	Interrupt vector index for a Supervisor Call exception
SYSTEM_INTERRUPT_PENDING_SV	Interrupt vector index for a Pending Supervisor interrupt
SYSTEM_INTERRUPT_SYSTICK	Interrupt vector index for a System Tick interrupt
SYSTEM_INTERRUPT_MODULE_PM	Interrupt vector index for a Power Manager peripheral interrupt
SYSTEM_INTERRUPT_MODULE_SYSCTRL	Interrupt vector index for a System Control peripheral interrupt
SYSTEM_INTERRUPT_MODULE_WDT	Interrupt vector index for a Watch Dog peripheral interrupt
SYSTEM_INTERRUPT_MODULE_RTC	Interrupt vector index for a Real Time Clock peripheral interrupt
SYSTEM_INTERRUPT_MODULE_EIC	Interrupt vector index for an External Interrupt peripheral interrupt
SYSTEM_INTERRUPT_MODULE_NVMCTRL	Interrupt vector index for a Non Volatile Memory Controller interrupt
SYSTEM_INTERRUPT_MODULE_DMA	Interrupt vector index for a Direct Memory Access interrupt
SYSTEM_INTERRUPT_MODULE_USB	Interrupt vector index for a Universal Serial Bus interrupt
SYSTEM_INTERRUPT_MODULE_EVSYS	Interrupt vector index for an Event System interrupt
SYSTEM_INTERRUPT_MODULE_SERCOMn	Interrupt vector index for a SERCOM peripheral interrupt. Each specific device may contain several SERCOM peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_SERCOM0`).

Enum value	Description
SYSTEM_INTERRUPT_MODULE_TCCn	Interrupt vector index for a Timer/Counter Control peripheral interrupt. Each specific device may contain several TCC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TCC0`).
SYSTEM_INTERRUPT_MODULE_TCn	Interrupt vector index for a Timer/Counter peripheral interrupt. Each specific device may contain several TC peripherals; each module instance will have its own entry in the table, with the instance number substituted for "n" in the entry name (e.g. `SYSTEM_INTERRUPT_MODULE_TC3`).
SYSTEM_INTERRUPT_MODULE_AC	Interrupt vector index for an Analog Comparator peripheral interrupt
SYSTEM_INTERRUPT_MODULE_ADC	Interrupt vector index for an Analog-to-Digital peripheral interrupt
SYSTEM_INTERRUPT_MODULE_PTC	Interrupt vector index for a Peripheral Touch Controller peripheral interrupt

8. Extra Information for SYSTEM INTERRUPT Driver

8.1. Acronyms

The table below presents the acronyms used in this module:

Acronym	Description
ISR	Interrupt Service Routine
NMI	Non-maskable Interrupt
SERCOM	Serial Communication Interface

8.2. Dependencies

This driver has the following dependencies:

- None

8.3. Errata

There are no errata related to this driver.

8.4. Module History

An overview of the module history is presented in the table below, with details on the enhancements and fixes made to the module since its first release. The current version of this corresponds to the newest version in the table.

Changelog
Initial Release

9. Examples for SYSTEM INTERRUPT Driver

This is a list of the available Quick Start guides (QSGs) and example applications for SAM System Interrupt (SYSTEM INTERRUPT) Driver. QSGs are simple examples with step-by-step instructions to configure and use this driver in a selection of use cases. Note that a QSG can be compiled as a standalone application or be added to the user application.

- Quick Start Guide for SYSTEM INTERRUPT - Critical Section Use Case
- Quick Start Guide for SYSTEM INTERRUPT - Enable Module Interrupt Use Case

9.1. Quick Start Guide for SYSTEM INTERRUPT - Critical Section Use Case

In this case we perform a critical piece of code, disabling all interrupts while a global shared flag is read. During the critical section, no interrupts may occur.

9.1.1. Setup

9.1.1.1. Prerequisites

There are no special setup requirements for this use-case.

9.1.2. Use Case

9.1.2.1. Code

Copy-paste the following code to your user application:

```
system_interrupt_enter_critical_section();

if (is_ready == true) {
    /* Do something in response to the global shared flag */
    is_ready = false;
}

system_interrupt_leave_critical_section();
```

9.1.2.2. Workflow

1. Enter a critical section to disable global interrupts.

   ```
   system_interrupt_enter_critical_section();
   ```

   Note: Critical sections *may* be nested if desired; if nested, global interrupts will only be re-enabled once the outer-most critical section has completed.
2. Check a global shared flag and perform a response. This code may be any critical code that requires exclusive access to all resources without the possibility of interruption.

   ```
   if (is_ready == true) {
       /* Do something in response to the global shared flag */
       is_ready = false;
   }
   ```
3. Exit the critical section to re-enable global interrupts.

   ```
   system_interrupt_leave_critical_section();
   ```

9.2. Quick Start Guide for SYSTEM INTERRUPT - Enable Module Interrupt Use Case

In this case we enable interrupt handling for a specific module, as well as enable interrupts globally for the device.

9.2.1. Setup

9.2.1.1. Prerequisites

There are no special setup requirements for this use-case.

9.2.2. Use Case

9.2.2.1. Code

Copy-paste the following code to your user application:

```
system_interrupt_enable(SYSTEM_INTERRUPT_MODULE_RTC);

system_interrupt_enable_global();
```

9.2.2.2. Workflow

1. Enable interrupt handling for the device's RTC peripheral.

   ```
   system_interrupt_enable(SYSTEM_INTERRUPT_MODULE_RTC);
   ```

2. Enable global interrupts, so that any enabled and active interrupt sources can trigger their respective handler functions.

   ```
   system_interrupt_enable_global();
   ```

10. Document Revision History

Doc. Rev.	Date	Comments
42122E	12/2015	Added support for SAM L21/L22, SAM DA1, SAM D09, and SAM C20/C21
42122D	12/2014	Added support for SAM R21 and SAM D10/D11
42122C	01/2014	Added support for SAM D21
42122B	06/2013	Corrected documentation typos
42122A	06/2013	Initial release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**